

Sérgio Matos<sup>1</sup> / Rui Antunes<sup>1</sup>

# Protein-Protein Interaction Article Classification Using a Convolutional Recurrent Neural Network with Pre-trained Word Embeddings

<sup>1</sup> DETI/IEETA, University of Aveiro, 3810-193 Aveiro, Portugal, E-mail: aleixomatos@ua.pt

## Abstract:

Curation of protein interactions from scientific articles is an important task, since interaction networks are essential for the understanding of biological processes associated with disease or pharmacological action for example. However, the increase in the number of publications that potentially contain relevant information turns this into a very challenging and expensive task. In this work we used a convolutional recurrent neural network for identifying relevant articles for extracting information regarding protein interactions. Using the BioCreative III Article Classification Task dataset, we achieved an area under the precision-recall curve of 0.715 and a Matthew's correlation coefficient of 0.600, which represents an improvement over previous works.

**Keywords:** Literature retrieval, protein-protein interactions, machine learning, recurrent neural networks, word embeddings

**DOI:** 10.1515/jib-2017-0055

**Received:** August 25, 2017; **Revised:** October 7, 2017; **Accepted:** November 3, 2017

## 1 Introduction

Extraction of protein-protein interaction (PPI) information from the literature is of utmost importance for biomedicine, since the understanding of disease, pharmacological and other processes requires the analysis of networks formed by these relations. Several databases maintain manually curated protein-protein interaction data but, since the primary source for identifying PPIs is the scientific literature, keeping these databases up-to-date is a demanding and expensive task. The use of named-entity recognition (NER) and relation extraction methods in assisted curation workflows has been shown to expedite this work [1], [2]. An important step in such workflows is document prioritization or triage, in order to select articles that are more likely to contain relevant information.

Retrieval and extraction of PPI related information has been a major focus of recent shared evaluations in the biomedical domain, namely in the BioCreative challenges. Lan et al. [3] compared the use of bag-of-words (BoW), interaction trigger words and protein named entities (NEs) features in a support vector machine (SVM) classifier, applied to the BioCreative-II PPI task data. Their best result, when using a single classifier, was obtained with a feature set containing BoW features and protein NEs co-occurring with interaction trigger words (F-score of 77 %). Abi-Haidar et al. [4] tested three classifiers in the same data set: SVM, variable trigonometric threshold classifier (VTT), and a nearest neighbor classifier with singular value decomposition (SVD) applied for feature selection. They reported a top F-score of 78 % using the VTT classifier with a feature set of 650 discriminating words.

Several other works have also addressed the problem of document prioritization for protein-protein interactions. Sumoela and Andrade [5] proposed a classification and ranking model to evaluate the entire MEDLINE database, the largest repository of scientific literature in the life sciences, with respect to any topic of interest. Their method is based on selecting words that commonly convey meaning, namely nouns, verbs, and adjectives, and relies on the different frequencies of these discriminating words between a set of relevant articles and a reference set. This approach is also behind the MedlineRanker web-service [6], which allows to retrieve a list of articles ranked by similarity to a training set defined by the user. One possibility, as referred by the authors, is to use a list of document identifiers obtained from a PPI database, therefore getting as result other articles related to that same topic. Marcotte et al. [7] proposed a log likelihood scoring function to identify articles discussing PPIs, using a feature set composed of 83 discriminating words selected from a training set of 260 MEDLINE

Sérgio Matos is the corresponding author.

© 2017, Sérgio Matos and Rui Antunes, published by De Gruyter.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 3.0 License.

abstracts involving yeast proteins. They reported an accuracy of 77 %, with a recall around 55 %, when articles with a log likelihood score of 5 or higher were selected.

The BioCreative III PPI Article Classification Task (ACT) counted with 52 submissions from ten participating teams [8]. Most teams applied some sort of machine learning technique, the best results being obtained using support vector machines, Maximum Entropy or Large Margin classifiers. The top performing teams used various levels of lexical analysis, including part-of-speech (PoS) tagging and named entity recognition (NER), and the best team overall also used dependency parsing to extract the textual features used for classification. Additionally, various teams used the manually assigned MeSH terms, which are indexing terms that provide information regarding the article's subject. The best AUC iP/R (area under the interpolated precision-recall curve) was 0.680 and the highest MCC (Matthew's correlation coefficient) was 0.553, with an accuracy of 89.2 % and an F-score of 61.4 % [9].

In recent works, deep learning classifiers have been applied successfully to many tasks, most notably when applied to image data but also in many text and natural language processing problems. Similarly to traditional neural networks, deep learning classifiers are composed of processing units arranged in data transformation layers that apply simple non-linear functions to obtain different levels of representation of the input data [10]. In deep learning, however, a larger number of layers and/or units per layer are used, which allows the method to learn more complex classification functions. Another great advantage of such methods is that they eliminate the feature engineering effort that is required in traditional machine learning [10].

For text based tasks, the input data needs to be encoded in a way that can be used by the deep network classifier. This can be achieved by representing each word as a vector of a relatively small dimension. This way, each document is represented by a sequence of word vectors which are fed directly to the network. Word embeddings is a technique that derives such vector representations of words from large unannotated corpora, representing words with similar semantics by vectors that are close to one another in the vector space [11]. The use of this representation together with deep learning techniques have led to improved results in different NLP tasks, including word sense disambiguation [12], text classification [13], and named entity recognition [14].

Convolutional neural networks (CNN) are one of the most popular network architectures used in deep learning, having been extensively applied in image recognition and classification problems with very good performance. Various works also demonstrate their application in text classification tasks [13], [15], [16]. Nonetheless, the sequential nature of natural texts can be better modeled by recurrent neural networks (RNN), which contain a feedback loop that allows the network to use information regarding the previous state. Long short-term memory (LSTM) networks are a special type of RNN in which a set of information gates is introduced that allow these networks to learn long-term dependencies while avoiding the vanishing gradient problem [17], [18], [19].

An important consideration when defining a deep neural network for a given classification problem is related to selecting the network topology, namely type and number of layers and number of units in each layer, and model parameters such as activation function in each layer, loss function and optimizer algorithm.

Another important aspect is related to overfitting, which means that the network is capable of learning the "best" representation for the data used in training but is not able to generalize to unseen data. Various strategies have been proposed and are commonly employed to address this problem, namely early stopping, dropout, and regularization. The first is based on stopping the training when the value of the loss function, measured in an held-out part of the training data, stops decreasing. Dropout freezes the weights of a fraction of the units in the previous layer. This means that after a training epoch the weights of those units are not changed, so that the network is forced to assign importance to other features and does not focus on some parts of the feature space. Regularization is a common strategy used for trying to avoid overfitting. We used L2 regularization, which increases the value of the loss function when larger weights are used and thus benefits the distribution of weights across more features.

In this work we applied deep learning methods, namely convolutional recurrent neural networks, for prioritization of MEDLINE articles containing protein-protein interaction information. The paper is organized as follows: the next section describes the methods and data used, followed by the presentation of results, and finally the conclusions.

## 2 Methods

This section describes the data and methods used. Text processing and classification tasks were implemented in Python, using the Scikit-learn machine-learning library [20] and Keras [21]. Word embedding models were calculated with the Word2Vec [22] implementation in the gensim framework [23].

## 2.1 Data

We used the dataset from the BioCreative III protein-protein interaction, article classification task (ACT) [8]. This corpus is composed of manually annotated MEDLINE abstracts, containing 2280 documents in the training set, 4000 in the development set, and 6000 in the test set. The training set has the same number of positive and negative examples, while the development and test sets are highly unbalanced, with around 15–17 % positive examples, reflecting the expected real scenario.

## 2.2 Model

The implemented network, illustrated in Figure 1, is composed of an embedding layer, a dropout layer, a convolutional layer followed by average pooling, a long-short term memory (LSTM) layer, and a dense layer with sigmoid activation function. The structure of the network was selected empirically by repeating various tests on the training and development sets.

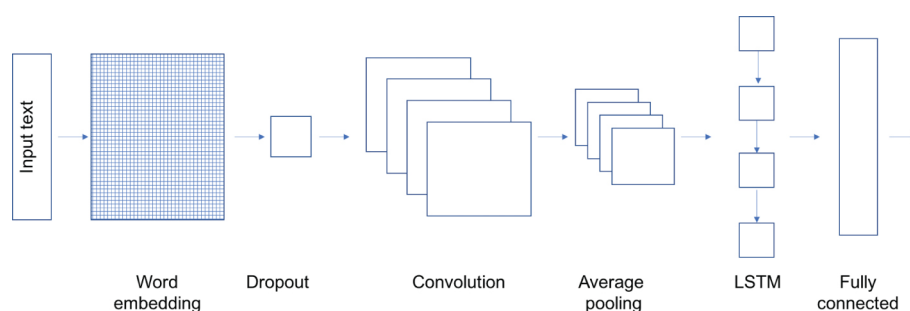


Figure 1: Deep network architecture.

The embedding layer represents the sequence of words in the documents by a sequence of embedding vectors. We used Word2Vec to create embedding models for 15 million abstracts from the full MEDLINE, containing around 775 thousand distinct words. We tested various models, with windows of 5, 20, and 50 and vector sizes of 100 and 300. Preliminary results showed that the embedding vectors of size 300 always gave better results. In the following experiments we used the model with 300 features calculated for window sizes of 20 and 50, as these gave consistently better results in preliminary tests than the other models.

The embedding layer is followed by a dropout layer in order to reduce over-fitting and thus improve generalization. The final network uses a dropout rate of 0.1.

A 1-dimensional convolutional layer is used to extract higher order features from the sequence of word vectors. We used a layer with 128 filters with kernel size set to 3 and rectified linear unit (ReLU) activation function. This is followed by average pooling over a window of 3.

The LSTM layer was configured with 128 units. Dropout was also used in this layer to prevent over-fitting, with a rate of 0.2. The output of the LSTM is then connected to the output layer. This consists of two neurons that apply a sigmoid activation function to obtain probability outputs in the range of 0–1 for each class. L2 regularization was used in this layer, with parameter set to 0.01.

Parameters such as the activation function in the convolution layer and in the final layer, number of convolution filters, the number of units in the LSTM, the dropout rates, regularization and optimizer function, were selected by performing a grid search over the parameters. For this, and to limit the computational cost, we selected 20 % of the development set (400 documents) and applied five-fold cross-validation for each parameter combination. In each fold, 80 % of the data were used for training and 20 % were used for evaluating the classification performance. In this process, we used a maximum of 10 training epochs with early stopping. Of the training data in each fold, 10 % were left out as validation data and the training was stopped if the loss in this validation data stopped decreasing.

Table 1 shows the parameter combinations tested, with the selected combination shown in bold.

Table 1: Network parameter combinations tested using grid search.

Parameter	Values
Number of units in LSTM	96, <b>128</b>
Number of filters in convolutional layer	96, <b>128</b>
Droupout rate after embedding layer	0, <b>0.1</b> , 0.2, 0.3, 0.4, 0.5

Droupout rate in LSTM	0, 0.1, <b>0.2</b> , 0.3, 0.4, 0.5
Activation in convolutional layer	<b>ReLU</b> , hyperbolic tangent ( <i>tanh</i> )
Activation in final layer	<b>sigmoid</b> , softmax
Optimizer algorithm	Adam, <b>RMSprop</b>
Regularization parameter	0, <b>0.01</b> , 0.02

Values in bold were selected as the best combination.

### 3 Results and Conclusions

Table 2 shows the results obtained on the test set of the BioCreative III article classification task. Two embedding models with different window sizes (20 and 50) were compared. The network parameters were set as described in the previous section.

**Table 2:** Evaluation results on the test set.

Word embeddings	Metrics			
	AUC PR	Acc.	MCC	P @ Full R
l = 300; w = 20	0.706	0.901	0.585	0.162
l = 300; w = 50	0.715	0.894	0.600	0.192

The two models with best results are shown. l: word2vec vector length; w: word2vec window size; AUC PR: Area under the precision/recall curve; Acc: Accuracy; MCC: Matthew's correlation coefficient; P@Full R: Precision at full recall.

For this task, the state-of-the-art results were achieved during the BioCreative III challenge. Comparing to the best results, our classifier shows an improvement in terms of area under the precision-recall curve (0.715 vs. 0.680) and Matthew's correlation coefficient (0.600 vs. 0.551), with a similar accuracy (0.894 vs. 0.892).

We present results for the prioritization of scientific articles containing information regarding protein-protein interactions, following a classification based ranking approach. We evaluated the combination of convolutional and recurrent neural networks, and show that these methods improve the classification and ranking performance over the previous state-of-the-art results. Importantly, these improvements were achieved without the need for features such as grammar relations extracted from full dependency parsing results, MeSH index terms, or the results of named-entity recognition systems.

On the other hand, setting of network parameters has an important impact on the final classification performance, but this can be achieved through grid search, as done in this work. The initialization of network weights may also have an impact, and weight initialization strategies are an interesting direction for improving the results of this approach.

### Acknowledgement

This work was supported by Portuguese National Funds through FCT - Foundation for Science and Technology, through a FCT Investigator grant (IF/01694/2013) and through Project IF/01694/2013/CP1162/CT0018.

**Conflict of interest statement:** Authors state no conflict of interest. All authors have read the journal's publication ethics and publication malpractice statement available at the journal's website and hereby confirm that they comply with all its parts applicable to the present scientific work.

### References

- [1] Krallinger M, Valencia A, Hirschman L. Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome Biol* 2008;9:1.
- [2] Wang Q, Abdul S, Almeida L, Ananiadou S, Balderas-Martínez YI, Batista-Navarro R. Overview of the interactive task in BioCreative V. Database. 2016;2016:baw119.

- [3] Lan M, Tan CL, Su J. Feature generation and representations for protein–protein interaction classification. *J Biomed Inform* 2009;42:866–72.
- [4] Abi-Haidar A, Kaur J, Maguitman A, Radivojac P, Rechtsteiner A, Verspoor K, et al. Uncovering protein interaction in abstracts and text using a novel linear model and word proximity networks. *Genome Biol* 2008;9:1.
- [5] Suomela BP, Andrade MA. Ranking the whole MEDLINE database according to a large training set using text indexing. *BMC Bioinformatics* 2005;6:1.
- [6] Fontaine JF, Barbosa-Silva A, Schaefer M, Huska MR, Muro EM, Andrade-Navarro MA. MedlineRanker: flexible ranking of biomedical literature. *Nucleic Acids Res* 2009;37(suppl 2):W141–6.
- [7] Marcotte EM, Xenarios I, Eisenberg D. Mining literature for protein–protein interactions. *Bioinformatics* 2001;17:359–63.
- [8] Krallinger M, Vazquez M, Leitner F, Salgado D, Chatr-aryamontri A, Winter A, et al. The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. *BMC Bioinformatics* 2011;12:1.
- [9] Kim S, Wilbur WJ. Classifying protein-protein interaction articles using word and syntactic features. *BMC Bioinformatics* 2011;12:1.
- [10] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521:436–44.
- [11] Bengio Y, Ducharme R, Vincent P, Jauvin C. A neural probabilistic language model. *J Mach Learn Res* 2003;3:1137–55.
- [12] Yepes AJ. Word embeddings and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation. *J Biomed Inform* 2017;73:137–47.
- [13] Kim Y. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar; Association for Computational Linguistics, 2014;1746–1751. DOI: 10.3115/v1/D14-1181.
- [14] Habibi M, Weber L, Neves M, Wiegandt DL, Leser U. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* 2017;33:i37–48.
- [15] Hu B, Lu Z, Li H, Chen Q. Convolutional neural network architectures for matching natural language sentences. In: Ghahramani Z, Welling M, Cortes C, Lawrence N.D, Weinberger K.Q, eds. *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 2. Cambridge: MIT Press, 2014:2042–2050.
- [16] Dos Santos CN, Gatti M. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, 2014:69–78.
- [17] Gers F.A., Schmidhuber J., Cummins F., et al.. Learning to forget: continual prediction with LSTM. *9th International Conference on Artificial Neural Networks: ICANN '99*. Edinburgh, UK; IET, 1999;850–855. DOI: 10.1049/cp:19991218.
- [18] Gers FA, Schmidhuber E. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Trans Neural Netw* 2001;12:1333–40.
- [19] Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling. In: *Thirteenth annual conference of the International Speech Communication Association*, 2012.
- [20] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;12:2825–30.
- [21] Chollet F. Keras. GitHub. 2015. <https://github.com/fchollet/keras>, accessed Jan 2017.
- [22] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, 2013:3111–3119.
- [23] Řehůřek R, Sojka P. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010:45–50.